



AFRL-RI-RS-TR-2010-195

## EXAMINATION OF MULTI-CORE ARCHITECTURES

---

*November 2010*

INTERIM TECHNICAL REPORT

*APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.*

STINFO COPY

**AIR FORCE RESEARCH LABORATORY  
INFORMATION DIRECTORATE**

## **NOTICE AND SIGNATURE PAGE**

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report was cleared for public release by the 88<sup>th</sup> ABW, Wright-Patterson AFB Public Affairs Office and is available to the general public, including foreign nationals. Copies may be obtained from the Defense Technical Information Center (DTIC) (<http://www.dtic.mil>).

AFRL-RI-RS-TR-2010-195 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE DIRECTOR:

/s/  
JOSEPH A. CAROLI  
Branch Chief

/s/  
EDWARD J. JONES, Deputy Chief  
Advanced Computing Division  
Information Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

**REPORT DOCUMENTATION PAGE***Form Approved*  
**OMB No. 0704-0188**

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington Headquarters Service, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC 20503.

**PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.****1. REPORT DATE (DD-MM-YYYY)**  
NOVEMBER 2010**2. REPORT TYPE**  
Interim Technical Report**3. DATES COVERED (From - To)**  
February 2010 – July 2010**4. TITLE AND SUBTITLE**

EXAMINATION OF MULTI-CORE ARCHITECTURES

**5a. CONTRACT NUMBER**

In House

**5b. GRANT NUMBER**

N/A

**5c. PROGRAM ELEMENT NUMBER****6. AUTHOR(S)**

Courtney Usmail

**5d. PROJECT NUMBER**

HPCC

**5e. TASK NUMBER**

IN

**5f. WORK UNIT NUMBER**

HO

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**Air Force Research Laboratory/RITB  
525 Brooks Road  
Rome, NY 13441-4505**8. PERFORMING ORGANIZATION  
REPORT NUMBER**

AFRL/RITB

**9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)**Air Force Research Laboratory/RITB  
525 Brooks Road  
Rome NY 13441-4505**10. SPONSOR/MONITOR'S ACRONYM(S)**  
AFRL/RITB**11. SPONSORING/MONITORING  
AGENCY REPORT NUMBER**  
AFRL-RI-RS-TR-2010-195**12. DISTRIBUTION AVAILABILITY STATEMENT**

Approved for Public Release; Distribution Unlimited. PA# 88 ABW-2010-5835.

Date Cleared: 2 November 2010.

**13. SUPPLEMENTARY NOTES**

**14. ABSTRACT** This multi-task research effort is designed to determine the applicability of various multi-core architectures to USAF warfighter needs. Furthermore, the resulting advancements in the area of high performance computing will benefit the mission areas of the Air Force Research Lab Information Directorate. This effort is broken down into four tasks; the first three tasks include investigating multi-core architectures, benchmarking and performance analysis techniques, and warfighter needs. The final task of the effort combines results from previous tasks and is the design and implementation of an experiment that will compare and contrast multi-core architectures based on their strengths, weaknesses, and applicability to US Air Force C4ISR needs. While there are a great number of multi-core architectures on the market, the first task of the effort aimed specifically at investigating architectures. The system specifications that were examined include the architecture design and performance specifications. For the architectures that were examined, the corresponding software development environment was also examined in greater detail; determining characteristics such as tool chains, programming models, and libraries. This technical memorandum serves as a point of reference for information on various multi-core architectures. The research for this report was done as part of an associated support task, JON HPCCINH1, under the RIDER project.

**15. SUBJECT TERMS**

Multi-Core Architectures, General Purpose Graphical Processing Units, High Performance Computing

**16. SECURITY CLASSIFICATION OF:****a. REPORT**  
U**b. ABSTRACT**  
U**c. THIS PAGE**  
U**17. LIMITATION OF  
ABSTRACT**

UU

**18. NUMBER  
OF PAGES**

18

**19a. NAME OF RESPONSIBLE PERSON**  
COURTNEY USMAIL**19b. TELEPHONE NUMBER (Include area code)**  
N/A

## TABLE OF CONTENTS

Section	Page
1.0 PROBLEM STATEMENT	1
2.0 BACKGROUND	1
3.0 ARCHITECTURE CHARACTERISTICS	3
3.1 NVIDIA Tesla	3
3.2 TILE64	4
3.3 Cell Broadband Engine	5
3.4 Tera-Op Reliable Intelligently Adaptive Processing System (TRIPS)	6
3.5 Intel	6
4.0 SOFTWARE DEVELOPMENT ENVIRONMENT	7
4.1 OpenCL	8
4.2 CUDA	8
4.3 Tiler MDE	8
4.4 TRIPS SDK	9
4.5 IBM SDK	9
5.0 SUMMARY	10
6.0 REFERENCES	11
7.0 APPENDIX	13

## LIST OF FIGURES

Figure		Page
1	Tesla Architecture	3
2	TILE64 Architecture	4
3	Single Tile Architecture	4
4	STI Cell Broadband Engine Architecture	5
5	TRIPS Architecture	6
6	Intel Corei7 Chip	6

## **1.0 PROBLEM STATEMENT**

The Air Force Research Laboratory Information Directorate has procured and installed a variety of multi-core architectures that will potentially enhance high performance computing needs of the US Air Force. In order to determine what military benefits may come from the application of high performance computing technology, it is essential to adequately understand multi-core architectures in greater detail. A survey was conducted to thoroughly assess the performance capabilities and intended applications of these architectures.

## **2.0 BACKGROUND**

In what has become known as “Moore’s Law,” Intel co-founder Gordon Moore predicted in 1965 that there would be an exponential growth in the number of transistors per circuit in microprocessor technology, approximately doubling every 24 months [1]. This growth in the number of transistors thereby leads to exponential growth in the performance of processors. Directly correlating to Moore’s Law, advancements in computer technology have driven the need for processors with greater speed, power, and overall performance. While Moore’s prediction has proved to be correct, several flaws exist with the method of exponentially increasing the number of transistors on a chip. One flaw is that processing speed is increasing at a much greater rate than memory speed, leading to increased memory latency. This is referred to as the memory wall [2]. For example, in the late 1980’s to 1990’s there were six to eight clocks per cycle required for memory access; current processors require over 200 clocks per cycle [1]. Another setback with increasing the number of transistors on a chip is the size of the transistor must significantly decrease in order to increase the number of transistors present. Respectively, the gate on the transistor that is responsible for switching the electricity on and off gets thinner, resulting in a decreased ability to block electron flow. Therefore, electricity is constantly being used and power is being wasted [3]. This is an example of the “power wall,” another performance problem regarding the excessive consumption of power [2]. This is the most prevalent problem which arises when the number of transistors on a chip increases, as there is a great increase in power density because every transistor requires power and generates heat [1]. Increasing the clock speed with the increased number of transistors is another cause of the increased power density. When the clock speed is increased the transistors switch faster and in turn use more power and create more heat [3]. Dissipation of the heat that is generated becomes a significant challenge. An additional problem encountered with single core processors is the “instruction-level parallelism wall” [2]. Instruction-level parallelism requires future instructions be determined prior to the completion or success of current instructions. Also, instruction-level parallelism requires additional safeguarding for instructions that are executed out of order [2].

To alleviate these problems, there has been a paradigm shift towards increasing the number of cores on a chip. The advantages of increasing the number of cores on a chip include: multiple cores can share components, perform a greater amount of work in a clock cycle, use less power, and have greater efficiency in multiple task processing. While single core chips have to assign time to work on individual programs, in turn causing errors and slow downs, multi-core processors are able to manage more work in parallel [1, 2]. In addition, doing more work per clock cycle allows the processor to run at a lower frequency, addressing the issue of power consumption and heat production [2]. With the development of multi-core processors comes the need for development of multithreaded software that can break up the applications and divide the work between cores using thread-level parallelism [1, 2]. Since the switch to increasing the number of cores rather than the number of transistors has been made, a vast number of multi-core processors have been developed to encompass a wide variety of performance characteristics.

While there exists a great many multi-core architectures, ranging from two to hundreds of cores on a chip, only a small portion of architectures were examined. There are several targeted applications for multi-core processors including general desktop computing, embedded computing, and high performance computing (HPC). The multi-core processors found in desktop computers have generally two to six identical cores and are not overly power efficient. Multicore processors used for embedded computing also have identical cores, but are designed with power consumption as a primary consideration, as they run on battery power. Finally, those multi-core processors which are used for high performance computing applications generally use many more cores than the desktop multi-core processors – up to 100 times as many cores [4].

**Table 1 – Representative List of Multi-Core Architectures by Application Domain [4]**

<b>General Purpose</b>	<b>HPC</b>	<b>Embedded</b>
AMD Phenom	AMD Radeon	ARM Cortex
Intel Core i7	Nvidia G200	XMOS XSI-G4
Sun Niagara	Intel Larrabee	Tensilica Xtensa
Intel Atom	IBM Cell	Tilera
TRIPS	Microsoft Xenon	

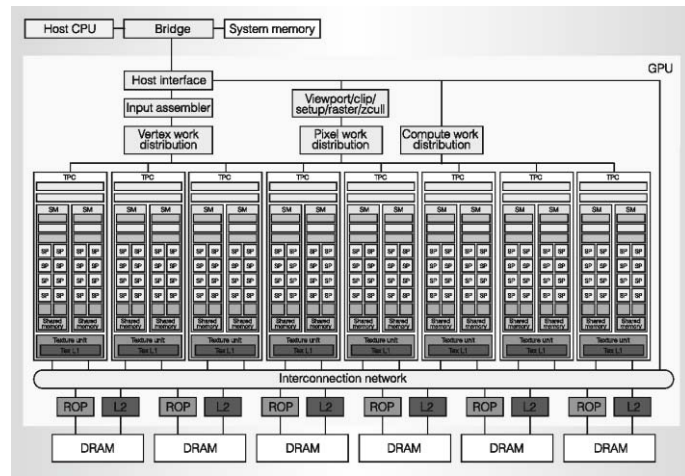
### 3.0 ARCHITECTURE CHARACTERISTICS

A literature survey was conducted to examine parameters such as number of cores, clock speed, and memory characteristics and ultimately determined the relative strengths and weaknesses of a chosen set of architectures. In addition, the on-chip set up of the architectures was also examined. The table in Appendix A summarizes the performance parameters for these architectures. Below is a more detailed overview of each architecture.

#### 3.1 NVIDIA Tesla

NVIDIAs Tesla graphical processing unit (GPU), shown in Figure 1, is based on a scalable processor array [5]. The architecture consists of streaming processor cores (SP), organized into streaming multiprocessors (SM) that are located in a maximum of eight independent processing units referred to as texture/processor clusters (TPC) [6]. The eight sub-units can be seen in Figure 1 illustrating the organizational hierarchy. Within every TPC there exists a texture unit which performs graphics calculations and is equipped with a Level 1 (L1) cache, and several streaming multiprocessors which are composed of a number of streaming processor cores [5].

Another feature of the Tesla GPU is the host interface, which is connected to the CPU and each TPC. The host interface is the means by which the CPU communicates with the GPU [6]. The host interface serves many functions such as thread management, data and instruction retrieval from the CPU, data retrieval from system memory, and context switching [5]. An additional feature is the Level 2 (L2) cache memory units which are connected to TPCs via an interconnection network. All calculations for the GPU are performed by the streaming multiprocessors. The streaming multiprocessors use groups of threads known as warps to manage program execution. Several warps may be run at the same time via interleaving instructions [5].



**Figure 1 – NVIDIA Tesla Architecture [5] © IEEE 2009**  
**Maciol et al. “A Survey of Multicore Processors: A**  
**Review of their Common Attributes.” *IEEE Signal***  
***Processing***

General purpose GPUs like the Tesla can be used for synthetic aperture radar, including geological mapping and disaster observation and management [7]. NVIDIA developed the



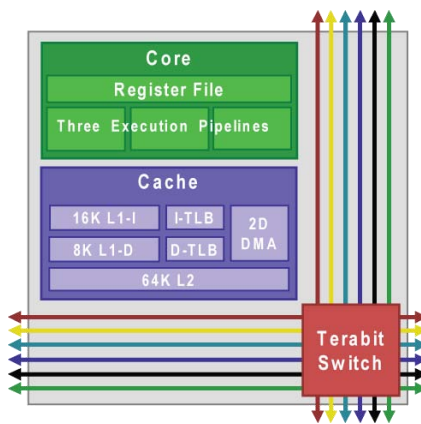
Compute Unified Device Architecture (CUDA) as the general-purpose programming model for the Tesla GPU family. This model uses standard C/C++ extensions and eliminates the need to translate scientific codes into graphics shading languages.

### 3.2 TILE64

The Tiler TILE64 architecture, shown in Figure 2<sup>1</sup>, consists of 64 tiled cores connected through an on-chip iMesh network [8]. The six mesh networks seen in Figure 3<sup>2</sup> connect the grid of tiles and have high-speed input/output (I/O) on the periphery with high-bandwidth and low-latency [8, 9]. Three networks are run by hardware and are responsible for moving data between tiles and memory. Three networks are for communication between cores and cores-I/O devices [8, 10].

A single tile consists of a processor core, cache system and switch as can be seen in Figure 3 [8].

Each tile can independently run an operating system

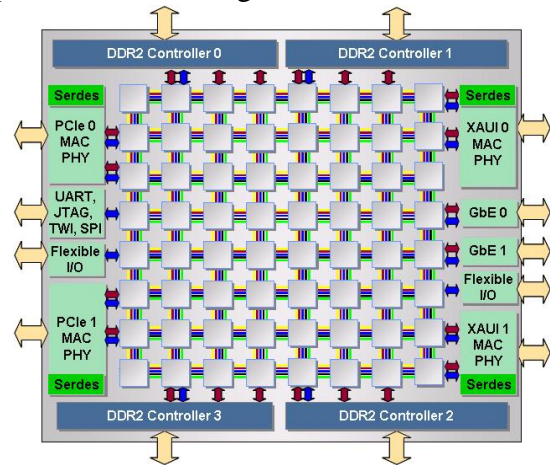


**Figure 3 – Single Tile Architecture [10]**

Each processor can be used for instructions for video and networking [10].

The cache system is a two-level hierarchy, where L1 is isolated from

complexity and L2 is shared. All 64 tiles are capable of viewing the L2 cache, therefore allowing it to function as a globally shared Level 3 (L3) cache [8]. The non-blocking switch connects the tile to the mesh and is composed of six independent mesh networks.



**Figure 2 – Tiler TILE64 Architecture [10]**

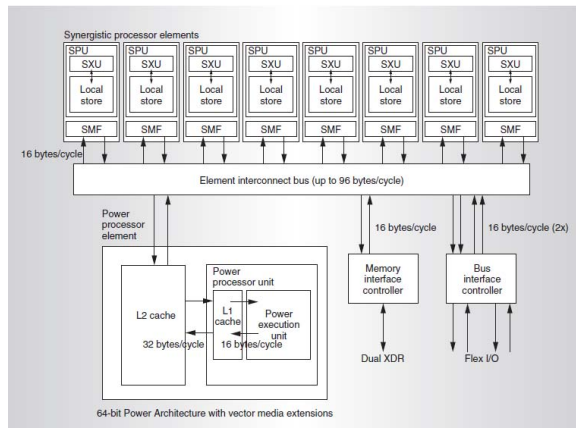
Intended applications of the TILE64 architecture include advanced networking, such as unified threat management and network monitoring, and digital video uses such as video on-demand servers and video surveillance [11]. Tiler Corporation developed its own real time software environment, the Multicore Development Environment (MDE) for developing on the TILE64 architecture. MDE is explained in greater detail under *Software Development Environments*.

<sup>1</sup> Image reprinted with permission from Tiler Corporation

<sup>2</sup> Image reprinted with permission from Tiler Corporation

### 3.3 Cell Broadband Engine

The Sony, Toshiba, and IBM (STI) Cell Broadband Engine, seen in Figure 4, is a heterogeneous multi-core architecture with nine cores: one Power Processing Element (PPE) and eight Synergistic Processing Elements (SPE) [12]. The PPE and SPE are connected through the Element Interconnect Bus (EIB) [13]. The PPE is based on the IBM Power Architecture processor with vector media extension and is built on a 64-bit PowerPC (PPU) and has two-levels of on-chip cache. The basic function of the PPE is to run the operating system and control the tasks [14]. The SPE is composed of a Synergistic Processing Unit (SPU), a Local Store, and a



**Figure 4 – STI Cell Broadband Engine Architecture [14] ©IEEE 2006**  
Gschwind et al. “Synergistic Processing in Cell’s Multicore Architecture” *IEEE Micro*

Memory Flow Controller (MFC) [12]. Each of the eight independent SPEs runs independent application threads. The SPE provides the majority of the compute performance for the Cell [14]. The EIB connects the SPEs, PPE, memory input control (MIC) and I/O controller [12].

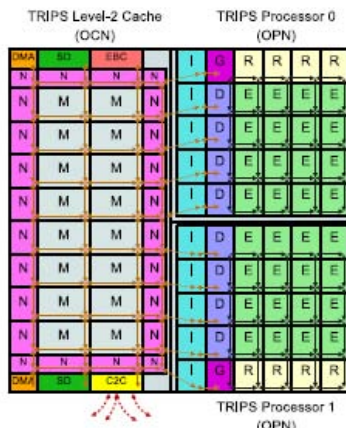
This architecture may be used for scalar codes in which the response time must be minimal. The Cell architecture also has increased support for applications which require both high computation requirements and high memory requirements such as video and image processing [13]. The software development environment used for programming the Cell is the IBM Cell Software Development

Kit [15]. The IBM SDK 3.0 will be explained in greater detail in the *Software Development Environments* section of this memorandum.

The Sony PlayStation3 (PS3) is fitted with the Cell BE processor. In the PS3, the Cell has only six of the eight SPEs available; one SPE is disabled at the hardware level and one SPE is set aside for the GameOS. The PS3 is also equipped with a dual-channelled Rambus *Extreme Data Rate* memory-system and provides 256 MB of memory, 200 MB of which is accessible to Linux and applications [15].

### 3.4 Tera-Op Reliable Intelligently Adaptive Processing System (TRIPS)

The TRIPS processor, seen in Figure 5, is a tiled architecture that contains two processor cores [16]. This architecture follows the design characteristics of grid processors and is block oriented. [17]. The two cores are divided up into five different types of tiles which are connected via seven micronets for tile communication, such as transferring data and routing control. Each tile can only interact with the tile directly neighboring it [16].



**Figure 5 – TRIPS Architecture [26]**

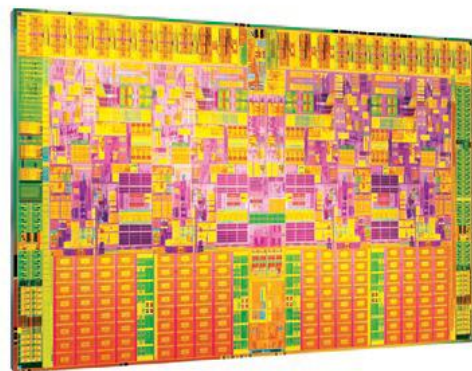
The processor cores implement the Explicit Data Graph Execution (EDGE) instruction set architecture and can be subdivided so they are able to run concurrent applications [17]. In addition to the two processor cores, each chip also consists of a secondary memory system that allows communication between the two cores. This memory system is organized into 16 tiles, each of which can be configured as L2 cache [16].

The TRIPS processor was designed under a Defense Advanced Research Projects Agency (DARPA) program titled

Polymorphous Computing Architectures (PCA), and can be used for general purpose computing [17]. Development for the PCA will be discussed in further detail in the following section.

### 3.5 Intel

The Intel Core i7 and Intel Xeon processors have several variations in performance and price. Both the i7 and Xeon processors with greater performance such as the i7-800 and Xeon 5500 series' are based on the Intel microarchitecture Nehalem [18, 19]. A general i7 processor based on this microarchitecture can be seen in Figure 6<sup>3</sup>. The Nehalem architecture has several key features unique to Intel architectures that enhance processor performance such as: Smart Cache Enhancements, QuickPath Technology, Turbo Boost Technology, Hyper-Threading Technology, Intelligent Power Technology, and Virtualization Technology. The largest component of the processor is the inclusive L3 cache through which the majority of data is passed.



**Figure 6 – Intel Core i7 Architecture [18]**

<sup>3</sup> Image reprinted with permission from Intel Corporation

The shared L3 cache is part of the Intel Smart Cache Enhancement and increases processor performance by decreasing latency and traffic among cores [18, 19]. Additionally, each core contains a smaller L1 and L2 cache. Another component integrated into each microprocessor is a memory controller. The memory controller and a high-speed interconnect included in the QuickPath Technology allows for a bandwidth 3.5 times greater to be delivered [19]. The Turbo Boost Technology makes it possible to increase the base frequency of the processor cores when requested by the operating system, or whenever necessary. Hyper-Threading Technology improves parallel computing capabilities by allowing simultaneous multithreading in each core [18,19]. The Nehalem architecture has an increased level of power efficiency due to the Intelligent Power Technology which essentially monitors the amount of power that is consumed. This makes it possible to decrease the power of the processor cores not being used to nearly zero [18]. The Intel Virtualization Technology boosts virtualization performance at the processor, chipset, and network adapter level [19].

The Core i7 is a general-purpose architecture, predominately used in high-end desktop computers [18]. The Intel Xeon processors are specially designed for several purposes such as standard enterprise servers, high-performance computing, and workstations [20]. Intel has developed a series of software development programs which can be used on the Intel processors, along with OpenCL.

#### **4.0 SOFTWARE DEVELOPMENT ENVIRONMENTS**

Software development environments provide the means by which programmers develop software for a given architecture. Software development environments can be vendor specific or general purpose. Greater performance capabilities are achieved with vendor specific software development environments, while general purpose software development environments provide greater ease of programming and portability. Software development environments may include several components such as libraries, simulators, integrated development environments, performance analyzers and toolchains.

## **4.1 OpenCL**

OpenCL (Open Computing Language) is a high-level language that can be used for parallel programming across heterogeneous platforms. OpenCL can be run on a variety of platforms; it is not restricted to one specific vendor [21]. The parallel programming model for OpenCL is based on C99. The OpenCL platform model consists of a host connected to one or more compute devices (OpenCL devices) which are composed of one or more compute units (cores) which are divided further into one or more processing elements [22]. The execution model in OpenCL consists of defining the problem domain and executing a kernel, a data or task parallel unit of executable code. Three main components make up OpenCL: platform layer, runtime and language [21]. OpenCL includes a long list of built-in functions, additional vectors types, work-items and workgroups [22].

## **4.2 CUDA**

The software platform developed by Nvidia to be used for parallel processing on NVIDIA GPU's is Compute Unified Device Architecture (CUDA). CUDA is vendor specific and allows general purpose computing on Nvidia GPU's [23]. There are four main components to the CUDA architecture which are: parallel compute engines within GPU's, OS kernel-level support, user-mode driver, and PTX instruction set architecture (ISA) [24]. CUDA uses extensions of C/C++ for the programming model, and also includes C/C++ software development tools and function libraries, all which make general purpose programming easier via code integration and type integration [23, 24].

## **4.3 Tilera MDE**

Tilera developed its own multicore development environment (MDE) to be used on their processors. Tilera's MDE has full support of the familiar C++ programming language, as well as standard C/C++ libraries [25]. The MDE can target a variety of applications including high performance computing and embedded computing using software environments such as SMP Linux and Bare Metal [10]. One important feature is the development environment allows for parallel debugging and parallel profiling which allows either control over the entire processor or individual cores [25]. It is also possible to view the processors activities and all statistics with the functional and source level profiling feature [10].

#### **4.4 TRIPS**

The DARPA PCA program developed its own software development environment to be used on the TRIPS architecture. The development environment includes “a compilation toolchain, a runtime system, debugging tools, and performance tools” [26]. The simulator’s and debugger developed for the TRIPS architecture have identical attributes to C++.

#### **4.5 IBM SDK for Multicore Acceleration**

The IBM SDK version 3.1 is the most current SDK used to program the STI Cell Broadband Engine. The SDK supports several languages including: C/C++, Assembler, Fortran, and ADA [27]. The development environment also includes GNU extended tools, IBM specialized compilers, standard libraries, an integrated development environment (IDE) and Full-System Simulator. Separate tools exist for programming processors units on the PPE and SPE’s. The SDK runs the most recent Linux kernel [28].

## 5.0 SUMMARY

An increasing drive for the development in multi-core architectures has been seen in recent years. Multi-core architectures are now developed by an abundance of vendors, in a wide range of designs, for several purposes. The interest in this paper was on multi-core architectures specifically used for high performance computing. Only a very small portion of the existing architectures were examined. Those that were examined are state of the art architectures procured by AFRL/RI. Among those discussed are general purpose GPUs (Nvidia Tesla), tiled or block architectures (Tilera TILE64 and TRIPS), heterogeneous architectures (Cell Broadband Engine), and general purpose architectures (Intel processors).

Several software development environments were also examined. Software development environments allow programmers to develop software for architectures and they can be vendor specific or general purpose. For example, NVIDIA developed its own software environment – CUDA-specifically for its GPGPU architectures such as the Tesla C1060, but also incorporate general purpose environments –OpenCL- that can be used across an array of architectures.

An examination of the various multi-core architectures provides information that can be used to assess the strengths and weaknesses of various architectures. Further research on available benchmarking techniques for multi-core architectures as well as the Air Force HPC requirements and applications will be done to assist in the overall goal of determining how specific multi-core architectures can fulfill the needs of the warfighter. The previous three tasks will be combined through experimentation to test the theoretical performance parameters and to compare and contrast their strengths, weaknesses, and applicability to US Air Force C4ISR applications.

An examination of the various multi-core architectures provides information that is useful in characterizing architectures, as well as assessing their strengths and weaknesses. Upon further research of available benchmarking techniques for multi-core HPC architectures and AF HPC applications an evaluation of how specific multi-core architectures can fulfill Air Force needs can be done.

## 6.0 REFERENCES

1. Koch, Geoff, "Intel's Road to Multi-Core Chip Architecture," Intel Corporation, 2006.
2. Manferdelli, John, Govinfaraju, Naga and Crall, Chris, "Challenges and Opportunities in Many-Core Computing," *Proceedings of the IEEE*, **96**, 5, 2008, pp 808-815.
3. Geer, David, "Chip Makers Turn to Multicore Processors", *IEEE Computer Society*, 2005, pp 11-13.
4. Blake, Geoffrey, Dreslinski, Ronald, and Mudge, Trevor, "A survey of Multicore Processors: A review of their common attributes", *IEEE Signal Processing Magazine*, 2009, pp 26-37.
5. Macioł, Pawel and Banaś, Krzysztof. "Testing Tesla Architecture for Scientific Computing: the Performance of Matrix-Vector Product." *Proceedings of the International Multiconference on Computer Science and Information Technology*, 2008, pp 285-291.
6. Lindholm, Erik, Nickolls, John, Oberman, Stuart and Montrym, John. "NVIDIA Tesla: A Unified Graphics and Computing Architecture," *IEEE Micro*, **28**, 2, 2008, pp 39-55.
7. Clemente, Carmine, di Bisceglie, Maurizio, Di Santo, Michele, Ranaldo, Nadia and Spinelli, Marcello "Processing of Synthetic Aperture Radar Data with GPGPU," *IEEE Workshop on Signal Processing Systems*, Finland, 2009, pp 309-314.
8. "The Five. Tile Processor Architecture Technology Brief." Tilera Corporation 2008. Dec 2009. <[http://tilera.com/pdf/ArchBrief\\_Arch\\_V1\\_Web.pdf](http://tilera.com/pdf/ArchBrief_Arch_V1_Web.pdf)>.
9. Bell, Shane, et al., "TILE64 Processor: A 64-Core SoC with Mesh Interconnect," *IEEE International Solid-State Circuits Conference*, 2008.
10. Agarwal, Anant (2009) *Tilera Technical Overview* [PowerPoint Slides].
11. "TILE64 Processor Product Brief", TILE64 Processor. 2008-2009. Tilera Corporation, Dec 2009. <[http://www.tilera.com/pdf/PB010\\_TILE64\\_Processor\\_A\\_v4.pdf](http://www.tilera.com/pdf/PB010_TILE64_Processor_A_v4.pdf)>.
12. Gschwind, Michael, Erb, David, Manning, Sid and Nutter, Mark, "An Open Source Environment for Cell Broadband Engine System Software" *Computer*, **40**, 6, 2007, pp 37-47.
13. Eichenberger, Alexandre, et al., "Optimizing Compiler for a CELL Processor." *14<sup>th</sup> International Conference on Parallel Architectures and Compilation Techniques*, Saint Louis, MO, 2005, pp 161-172.
14. Gschwind, Michael et al., "Synergistic Processing in Cell's Multicore Architecture." *IEEE Micro*, **26**, 2, 2006, pp 10-24.
15. Buttari, Alfredo, Dongarra, Jack, Kursak, Jakub, *Limitations of the PlayStation 3 for High Performance Cluster Computing*, LAPACK Working Note 185, Technical report ut-cs-07-598, UT Knoxville, 2007.
16. Sankaralingam, Karthikeyan, Nagarajan, Ramadass, McDonald, Robert et al., "Distributed Microarchitecture Protocols in the TRIPS Prototype Processor," *39<sup>th</sup> Annual International Symposium on Microarchitecture*, Orlando, FL, 2009.
17. Sankaralingam, Karthikeyan et al., "TRIPS: A Polymorphous Architecture



- for Exploiting ILP, TLP, and DLP” *ACM SIGARCH Computer Architecture News*, **31**, 2, May 2003, pp. 422-433.
18. Casazza, Jeff, “Intel Core i7-800 Processor Series and the Intel Core i5-700 Processor Series Based on Intel Microarchitecture (Nehalem),” *Intel Corporation*, 2009.
  19. Casazza, Jeff, “First the Tick, Now the Tock: Intel Microarchitecture (Nehalem),” *Intel Corporation*, 2009.
  20. *Intel Corporation*, “Intel Xeon Processor 5400 Series,” 2009.
  21. Zhang, Wenyu, Zhang, Li, Sun, Shangmin, Xing, Yuxiang, Wang, Yaj, “A Preliminary Study of OpenCL for Accelerating CT Reconstruction and Image Recognition,” *2009 IEEE Nuclear Science Symposium Conference Record*, 2009.
  22. “OpenCL Parallel Computing for Heterogeneous Devices,” *Khronos Group*, 2009.
  23. Halfhill, Tom, “Parallel Processing With Cuda. Nvidia’s High-Performance Computing Platform Uses Massive Multithreading,” *Microprocessor Report*, 2008.
  24. “NVIDIA CUDA Architecture: Introduction and Overview.” NVIDIA GPU Computing Developers Home Page. April 2009. NVIDIA Corporation. Jan 2010.  
<[http://developer.download.nvidia.com/compute/cuda/docs/CUDA\\_Architecture\\_Overview.pdf](http://developer.download.nvidia.com/compute/cuda/docs/CUDA_Architecture_Overview.pdf)>.
  25. Tilera Corporation, Multicore Development Environment Product Brief, [www.tilera.com](http://www.tilera.com)
  26. Keckler, Stephen, Buger, Doug, McKinley, Kathryn, Crago, Steve, Lethin, Richard, *Tera-Op Reliable Intelligently Adaptive Processing System (TRIPS) Implementation*, AFRL-RI-WP-TR-2008-1529, The University of Texas at Austin, Sep 2008.
  27. “Software Development Kit for Multicore Architecture Version 3.1 Programmer’s Guide,” Cell Broadband Engine Resource Center. 2008. IBM Corporation. Feb 2010.  
<[http://public.dhe.ibm.com/software/dw/cell/CBE\\_Programmers\\_Guide\\_v3.1.pdf](http://public.dhe.ibm.com/software/dw/cell/CBE_Programmers_Guide_v3.1.pdf)>.
  28. Maximo, Andre, Farias, Ricardo, Cox, Guilherme, Bentes, Cristiana, “Unleashing the Power of the Playstation 3 to Boost Graphics Programming,” *XXII Brazilian Symposium on Computer Graphics and Image Processing*, Rio de Janeiro, Brazil, 2009.
  29. “Board SpecificationL Tesla C1060 Computing Processor Board”, January 2009. NVIDIA Corporation. Nov 2009.  
<[http://www.nvidia.com/docs/IO/56483/Tesla\\_C1060\\_boardSpec\\_v03.pdf](http://www.nvidia.com/docs/IO/56483/Tesla_C1060_boardSpec_v03.pdf)>.
  30. “Board Specification Tesla C2050 and Tesla C2070 Computing Processor Board”, July 2010. NVIDIA Corporation. July 2010. <[http://www.nvidia.com/docs/IO/43395/BD-04983-001\\_v03.pdf](http://www.nvidia.com/docs/IO/43395/BD-04983-001_v03.pdf)>.
  31. “Intel Core i7-960 Processor Specifications,” Intel Corei7 Desktop Processors. Intel Corporation. Feb 2010. <<http://ark.intel.com/Product.aspx?id=37151>>.

## 7.0 APPENDIX

### Performance Capabilities

Architecture	Tesla C1060 [29]	Tesla C2050 [30]	TILE64 [9, 10, 11]	CELL [12, 13]	Intel Core i7- 960 [31]	Intel Xeon X5450 [20]	TRIPS [15, 16]
# of Cores	240	448	64	9 (PS3 7)	4	4	4
Clock Speed	1.3 GHz	1.25-1.4 GHz	700-866 MHz	3.2 GHz (PS3 3.2)	2.8-3.46 GHz	3 GHz	2.8-3.4 GHz
Total Memory	4 GB	3-6 GB	4 GB	256 MB	Unknown	4 GB	Unknown
Memory BW	102GB/s	170 GB/s	25 GB/S	25.6 GB/S	25.6 GB/s	32 GB/s	10.5 GB/s
Memory Speed	800 MHz	1.8-2.0 GHz	800 MHz	Unknown	1066 MHz	667 MHz	1.8 GHz
Memory Interface	512-BIT	384-BIT	[4] 72-BIT	128-BIT	192-BIT	128-BIT	[4] 64-BIT
Cache	Unknown	1 MB L1 768 KB L2	16 KB L1 64 KB L2 DIST L3	32 KB L1, 512 KB L2 256 KB LS	64 KB L1, 8 MB L3	64 KB L1 12 MB L2	4 MB L2
Peak Performance GFLOPS/S	SP-933	SP-1880 DP- 600	443 BOPS	SP- 204.8 PS3-153 DP- 14.6	SP: 51.2	SP: 96	Unknown